COMP42315 ASSIGNMENT

WEB SCRAPING, DATA ANALYSIS AND VISUALISATION

1. URL CRAWLER

To crawl all URLs a BeatifulSoup object (BeautifulSoup library) is created with two arguments: the HTML text the object is based on, and the parser used to convert the HTML text into an object.[1] This BeautifulSoup object represents the document as a nested data structure and can easily be navigated over. In this assignment, the body is crawled as it holds the bulk of the webpage's textual content. URL's are extracted from divs with the class: "ImgIconPublicationDiv". This is a robust solution as publication links are likely to stay embedded in the publication icons, even if the website is updated. After locating the image div, the ".find" function looks for "a", indicating the beginning of a href tag. This process is repeated using a for loop. All URLs are stored in a final list. Subsequently, the URLs are converted into global URLs by appending information such as the host name.

2. CRAWLING TEXT BASED INFORMATION

Crawling all text-based information involved finding HTML elements or CSS attributes which hold the desired information. For instance, in this HTML script only publication titles are given H1 tags. Therefore, the title was efficiently extracted using H1 tags and specifying that only text should be returned (html → body → h1.text). When the tag was not unique, CSS attributes were necessary to locate specific information. For example, the date and abstract were retrieved using "attrs = {"style":"

In most cases, information had to be separated from chunks of information also retrieved by the same tag. In such cases, it was efficient to utilise the library RegEx – regular expressions operations. This allowed string information to be separated based on specific search patterns.[2] Implementing RegEx within an if else clause, iteratively found the beginning and ending of the desired information. For example, when extracting the citation, RegEx found each beginning with "Citation:" and ending with "##". Afterwards, the string containing citation information was spilt and only the desired information stored in a list. If no citation was found, it was appropriate to append NA.

In some cases, the data needed further munging. Splicing, splitting, and replacing parts of returned information were a simple and effective solutions. For example, in the list of authors ".replace" was used to replace "and" with a comma. This allowed author names to be isolated by splitting them by each comma.

3. TEXTUAL ANAYLSIS: FINDING 100 MOST POPULAR WORDS

Finding the 100 most popular words involved parsing each page and extracting the title and abstract with the ".find" argument. Subsequently, the titles and abstracts were pre-processed so the computer could comprehend them later. This involved converting all characters to lower case, ensuring the program does not differentiate capitalised words. Additionally, RegEx was utilised to remove all punctuation. This pre-processed data was compiled into one list.

The following stage converted the list into one string by joining each space. Importantly, as the computer now reads the titles and abstract a string, textual data munging can be done. Here, the notion of stop words is useful. Stop words are highly frequent words with little semantic relevance.[3] To yield a list of meaningful words, stop words are removed from the string. In this assignment, some stop

[1] Mitchell, (2018), 9.

[2] Python RegEx. W3Schools. Website.

[3] Bird, et al. (2009) 60.

words are frequent within academic English, yet are semantically irrelevant, such as ‘demonstrate.’ Additionally, grammatically necessary but semantically meaningless words are removed. Examples include the article ‘the’ and modal verb ‘should’. Removing meaningless words results in a body of words that are recognisably relevant to academia and computer science. This algorithm is effective as it provides a simple text filtering solution without importing external text analysis libraries. (Figure 1) The string containing titles and abstracts without stop words was then spilt into individual words. Using counter from collections, the 100 most common words were counted.

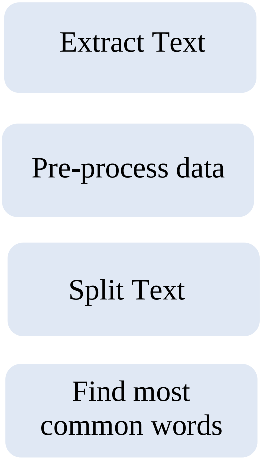

Figure 1. Flow chart of the algorithm to extract 100 most popular words.

4. DATA ANALYSIS AND VISUALISATION: AUTHOR COLLABORATIONS

Analysing author collaborations requires a data frame comprised of a column for paper titles and a column for each author. A dictionary initialises the data frame with the column name and type. Then using key value pair mapping, the data frame is filled with key value pairs. The title is added first and then a for loop adds each string to a new value to the object. Consequently, the data frame is filled up systematically. Using a nested for loop and the function .iterrows, each author is iteratively made into a column name. Each entry in the data frame is initially set to 0 and switched to one where an author is present in the paper title.

With this data frame, author collaborations are visualised using the correlation matrix heatmap in the Seaborn library. (Figure 2) A diverging colour palette makes this visualisation easy to read. Places where authors collaborate or correlate, appear dark. A numeric scale is also provided, with 1 being the strongest correlation. This heatmap shows every author collaborates with Hubert P.H Shum.

In addition, the authors Edmond S. L. Ho and Naoya Iwamoto frequently collaborate with other authors. Author collaborations are visualised using a single correlation map, where minus values indicate no collaboration. For instance, figure 3 shows all author correlations with Naoya Iwamoto as the dependant variable in a descending manner. Iwamoto collaborates with Shigeo Morishima and Hubert P.H Shum often, and with Wakana Asahina less often. Similarly, figure 4 shows the author collaborations with. Edmond S. L. Ho.

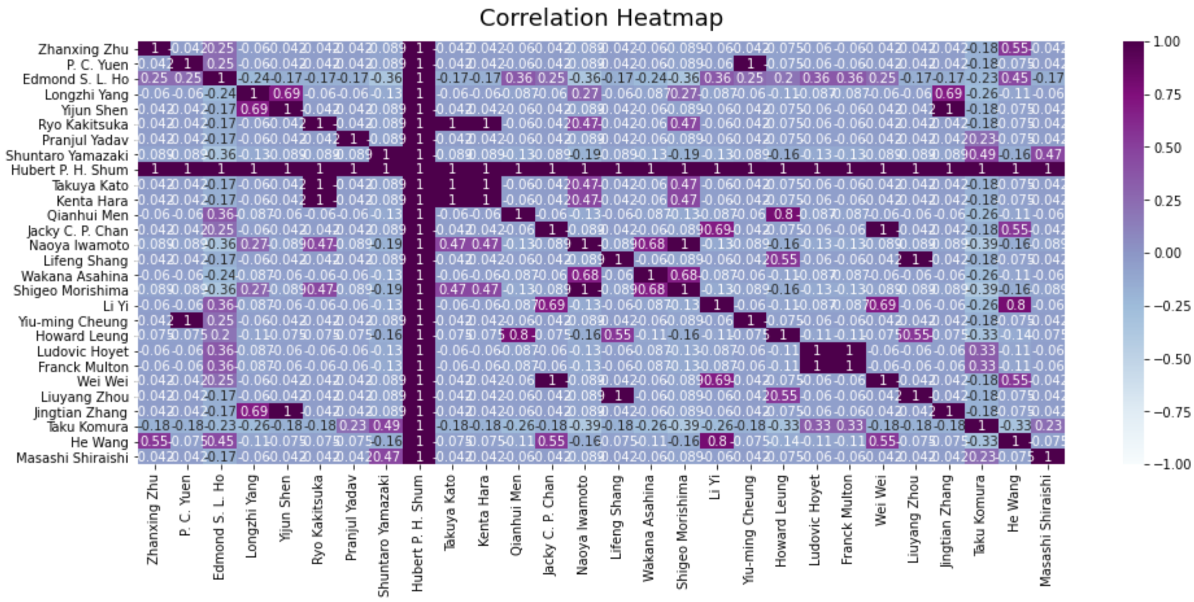

Figure 2. Correlation heatmap of correlations between authors.

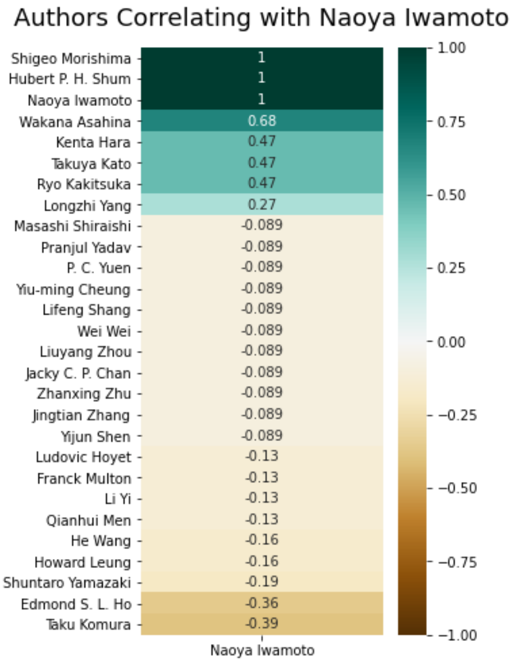

Figure 3. Response variable heatmap. Author correlations with Naoya Iwamoto.

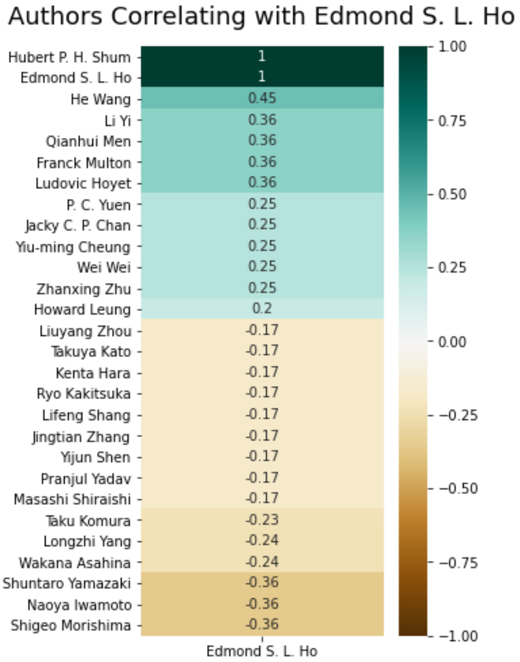

Figure 4. Response variable heatmap. Author correlations with Edmond S. L. Ho.

5. DATA ANALYSIS AND VISUALISATION: CITATION ANAYLSIS

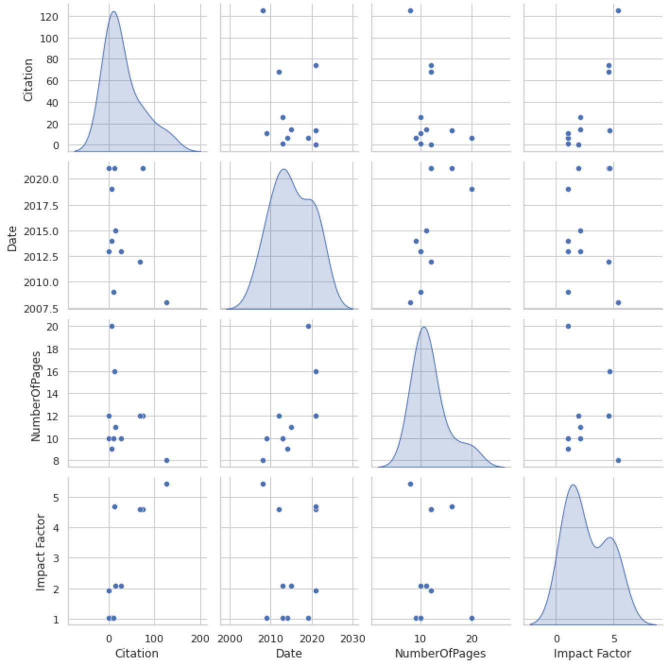

Figure 5. Pairs plot. Numeric features correlating with "Citation".

Visualising and analysing how features of a publication affect its citation involves creating a data frame which stores data extracted by the web crawler. Missing values were dropped from the data frame, while any numbers automatically set to floats were converted to integers.

In the analysis, pairs plots usefully highlight relationships between "Citation" and numeric factors. (Figure 5) The features "Date" and "Impact factor" positively relate to "Citation". The significance of features is explored in greater depth with a correlation map. (Figure 6) In this correlation map "Citation" is the dependant variable and analysed against the predictors: impact factor, number of pages, date, and titles. Publication titles are one-hot-encoded and treated as separate variables. This plot shows "Impact factor" correlates the most with "Citation", since it is a measure of how frequently a journal is cited. Interestingly, the paper "Interaction Patches for Multi-Character Animation" is the most significant feature affecting citation. This indicates paper popularity is the highest predictor of citation frequency.

Interestingly, "Number of Pages" and "Date" negatively correlate with "Citation". The relationship between "Number of Pages" , "Date" and "Citation" is further explored using linear regression. Linear regression models are created using the LinearRegression functionality from sci-kit learn. The x and y variables are defined, and the data is separated into training and testing sets. Using the random_state=42 argument ensures the data set is reproducible, meaning the program shuffles the data in the same way each time.

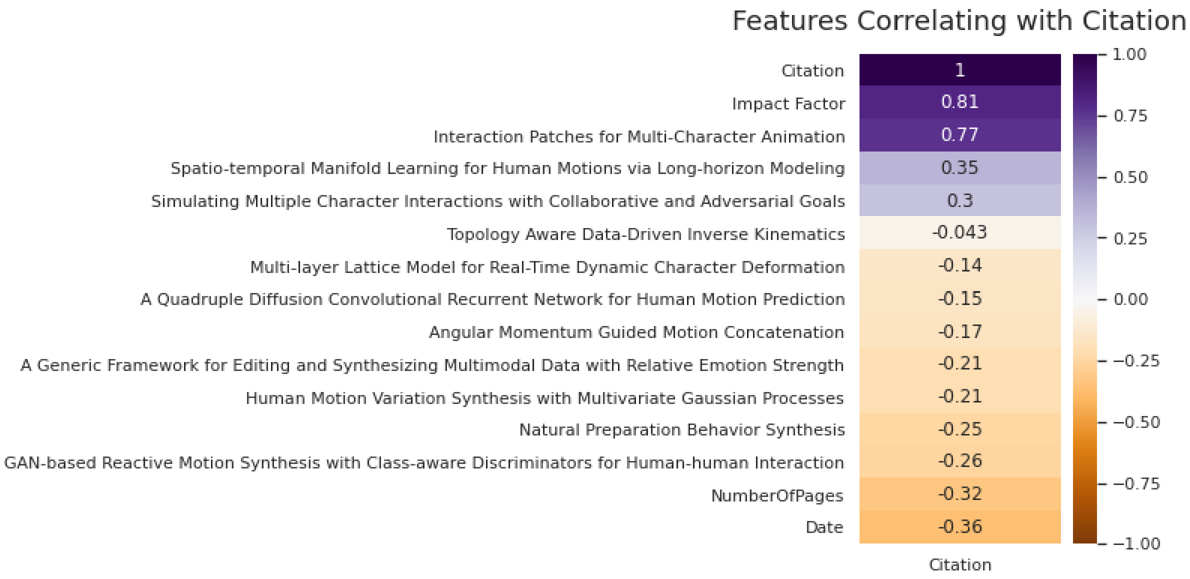

Figure 6. Response variable heatmap. Features correlating with “Citation”.

The scatter plot detailing date against citation shows the regression model predicts a strong negative association between “Citation” and “Date”. (Figure 7) Consequently, it is observed more recent papers have fewer citations it has. However, it must be noted that the training data (in blue) is most likely too sparse to form reliable and valid predictions. This is reflected by the R^2 value of 0.59, indicating the goodness of fit for this model is low.

Similarly, the R^2 value for the regression model predicting “Number of Pages” is extremely low. (Figure 8) This indicates “Page Number” is not a strong indicator of whether a paper is cited frequently.

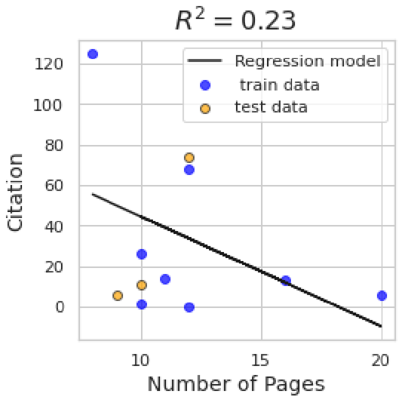

Figure 8. Scatter plot showing the regression model predicting “Citation” with “Number of Pages” as the predictor.

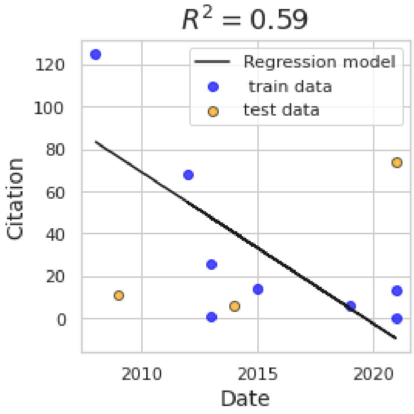

Figure 7. Scatter plot showing the regression model predicting “Citation” with “Date” as the predictor.

To visualise how “Publishers” affect “Citation” a catplot is used. (Figure 9) The four publishers are plotted with their corresponding citation values. The plot indicates IEEE and ACM are most frequently cited publishers. However, there is not enough data to draw a conclusive correlation between “Publisher” and “Citation”.

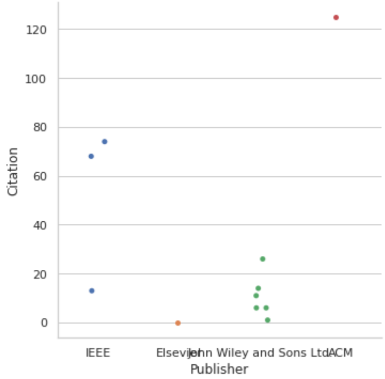

Figure 9. Catplot showing the categorial date “Publisher” plotted against “Citation”.

Bibliography

Beazley, D and Jones, B (2013) Python Cookbook, O'Reilly, Sebastopol

Bird, S; Klein, E; and Loper, E (2009) Natural Language Processing with Python, O'Reilly, Sebastopol

Géron, A (2019) Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly, Sebastopol

Matthes. E (2019) Python Crash Course, San Francisco

McKinney, W (2018) Python for Data Analysis, O'Reilly, Sebastopol

Mitchell, R (2019) Web Scraping with Python, O'Reilly, Sebastopol

Pandas "pandas.DataFrame.iterrows" <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.iterrows.html> [Last accessed: 16 February 2022]

Seaborn "Plotting with Categorical Data" < https://seaborn.pydata.org/tutorial/categorical.html> [Last accessed: 16 February 2022]

W3Schools "Python RegEx" <https://www.w3schools.com/python/python_regex.asp> [Last accessed: 16 February 2022]